

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Space-filling Curves in Generating Equidistributed Sequences and Their Properties in Sampling of Images

Ewa Skubalska-Rafajłowicz and Ewaryst Rafajłowicz

Institute of Computer Eng., Control and Robotics,

*Wrocław University of Technology Wybrzeże Wyspiańskiego 27, 50 370, Wrocław
Poland*

1. Introduction

Intensive streams of video sequences arise more and more frequently in monitoring the quality of production processes. Such streams not only have to be processed on-line, but also stored in order to document production quality and to investigate possible causes of insufficient quality. Direct storage of a video stream, coming with the intensity 10-30 frames per second with a resolution of 1-8 megapixels, from one production month would require 100-500 terra bytes of a disk (or tape) space. A common remedy is to apply compression algorithms (like MPEG or H264), but compression algorithms usually introduce changes in gray-levels or colors, which is undesirable from the point of view of identifying defects and their causes.

For these reasons we return to the traditional idea of sampling images, followed by loss-less compression. However, classical sampling on a rectangular grid is insufficient for our purposes, since it is still too demanding from the point of view of storage capacity. Our experience of using equidistributed (or quasirandom) sequences as experimental sites in non-parametric regression function estimation Rafajłowicz and Schwabe (2003); Rafajłowicz and Schwabe (2006); Rafajłowicz and Skubalska-Rafajłowicz (2003) suggests that such sequences can be good candidates for sampling sites. Roughly speaking, the reason is in that the projection of a 100×100 rectangular grid on the axes has 100 points, while a typical equidistributed sequence of the length 10^4 provides again 10^4 points when projected onto the same axes. The idea of using equidistributed (EQD) sequences in sampling images was firstly described in Thevenaz (2008), where it was used for image registration. Our goals are different and we need more specialized sampling schemes than a "general purpose" Halton's sequence, which was used in Thevenaz (2008).

Our aim is to propose a new method of generating equidistributed sequences, which is based on space-filling curves. Due to the remarkable properties of space-filling curves (SFC), which preserve volumes and (to some extent) neighborhoods, the proposed sequences are well-suited for sampling of images in such a way that samples can be processed similarly as an original image. We concentrate mainly on 2D images here, but 3D images are also covered by the theoretical properties. Simple reconstruction schemes, which are well-suited for industrial images, are also briefly discussed. We also indicate ways of generating sampling sequences

and reconstructing underlying images by neural networks, which are based on weighted averaging of gray-levels of nearest neighbors.

Let us note that space-filling curves have been used in image processing for image compression Kamata et al (1996); Lempel and Ziv (1986); Schuster and Katsaggelos (1997); Skubalska-Rafajłowicz (2001b), dithering Zhang (1998); Zhang (1997) halftoning Zhang and Webber (1993) and median filtering Regazzoni and Teschioni (1997); Krzyżak (2001). However, the measure and neighborhoods-preserving properties of these curves were not fully exploited.

The chapter is organized as follows.

1. In Section 2 we collect some known and certain not so well-known properties of space-filling curves, including the Hilbert, the Peano and the Sierpiński curves. In addition to measure-preserving properties, we provide an efficient algorithms for calculating approximations to selected space-filling curves. The definition and elementary properties of equidistributed sequences are recalled at the end of Section 2 with the emphasis on the Weyl sequences, which are used as the building block in the rest of the chapter.
2. The proposed way of generating equidistributed sequences is presented in Section 3. It is based on transforming the Weyl one-dimensional sequence $t_i = \text{fractionalpart}(i\theta)$, $i = 1, 2, \dots, \theta - \text{irrational}$, by a space-filling curve. We shall prove that sequences generated in this way are also equidistributed. The choice of θ is crucial for the practical behavior of the sampling scheme. Roughly speaking, θ should be an irrational number, which approximates badly by rational numbers.
3. In Section 4 we discuss some properties of our equidistributed sequences as a sampling scheme for 2D images.
 - We shall prove that the spectrum of a wide class of images can be reconstructed from samples when their number grows to infinity. By "wide class" we mean measurable functions, which allow for discontinuities.
 - We exploit the measure-preserving properties of space-filling curves in order to show that moments of images can easily be approximated from samples.
 - It will also be shown how simple image processing tasks can be performed, utilizing natural ordering of samples, which preserves neighbors in an image.
4. In section 5 we discuss two algorithms for the approximate reconstruction of the underlying image from samples. The first is based on the inversion of the spectrum estimate and it can be used for one image. The second one is based on the nearest neighbor (NN) technique, but it can be speeded up by preprocessing and storing (NN) addresses. This technique is useless for one image, but it is valuable when one needs to store a very long video sequence without degradation of pixel values, since NN addresses use only a very small portion of storage memory, while we gain on the reconstruction speed. The next reconstruction scheme, which is proposed here is based on neural networks of the radial-basis functions (RBF) type. We shall also provide the examples of sampling, processing and reconstructing industrial images.

2. Preliminaries

Our aim in this section is to collect known facts concerning space-filling curves and quasi-random sequences, which are useful for explaining the proposed way of sampling.

2.1 Space-filling curves – basic facts

In the 19th and at the beginning of the 20th century, space-filling curves were developed and investigated as mathematical "monsters", since they are continuous, but nowhere differentiable.

2.1.1 Definition

From those pioneering times researches more frequently treat space-filling curves as useful tools. The first applications were in approximate, multidimensional integration, see, e.g., Kuipers and Niederreiter (1974). The next area where they happened to be useful is scanning images Lamarque and Robert (1996); Cohen et al (2007) and the bibliography cited therein. Note that scanning images by a space-filling curve is the task, which is different from our goals, since the curve is expected to visit all the pixels in an image. Thus, scanning along a space-filling curve provides only linear ordering of pixels. Furthermore, in the above-mentioned papers additional features of space-filling curves, such as their ability to preserve closeness or area, were not used. Scanning images with utilization of some properties of space-filling curves for estimating the median was proposed in Krzyżak (2001). One more area of applications was proposed in Skubalska-Rafajłowicz (2001a), where space-filling curves were used as a tool in the Bayesian pattern recognition problems.

Definition 1. A space-filling curve is a continuous mapping $\Phi : I_1 \xrightarrow{\text{onto}} I_d$, where $I_d \stackrel{\text{def}}{=} [0, 1]^d$ is d -dimensional unit cube (or interval $I_1 = [0, 1]$), $d \geq 1$.

We cannot draw a space-filling curve, since it maps $[0, 1]$ onto I_2 . Thus, the image of I_1 by Φ would be completely black in the unit square. However, we can draw an approximation to such a curve, as is illustrated in Fig. 1.

It is important to mention that these curves can be approximated to the desired accuracy by implementable algorithms (see below).

The well-known curves constructed by Hilbert, Peano and Sierpiński possess properties Sagan (1994); Milne (1980); Moore (1900); Sierpiński (1912); Platzman and Bartholdi (1989); Skubalska-Rafajłowicz (2001a), which are stated in the two next subsections. These properties are stated for $d = 2$, but they holds for $d > 2$ with obvious changes.

2.1.2 Most important properties

The formula for changing variables in integrals, which is stated below, was used for constructing multidimensional quadratures. Here, we shall need it for approximating the Fourier spectrum of images from samples.

Property 1 (F1 – Change of variables). Let $\Phi : I_1 \xrightarrow{\text{onto}} I_d$ be a space-filling curve. Then, for every measurable function $g : I_2 \rightarrow R$

$$\int_{I_2} g(x) dx = \int_0^1 g(\Phi(t)) dt, \quad (1)$$

where $x = [x^{(1)}, x^{(2)}]^T$ and T denotes the transposition and the integrals in (1) are understood in the Lebesgue sense.

The Lipschitz continuity of the curves constructed by Hilbert, Sierpiński and Peano is somewhat more demanding property, than the continuity required in the above definition, but is less than necessary for the first order differentiability.

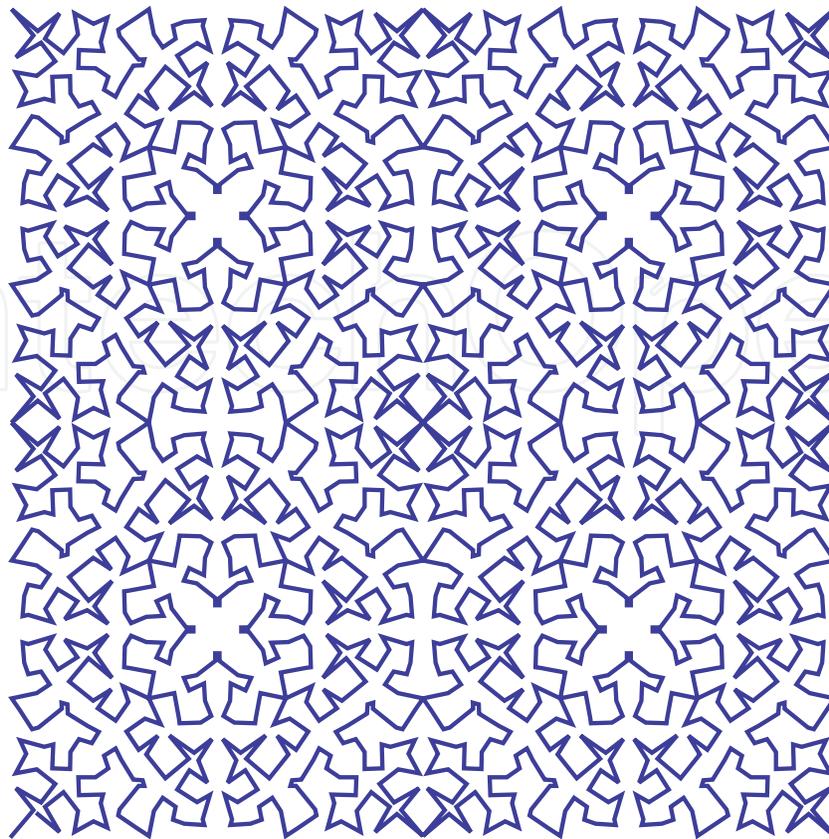


Fig. 1. An approximation to the Sierpiński SFC.

Property 2 (F2 – Lipschitz continuity). *There exists $C_\Phi > 0$ such that*

$$\|\Phi(t) - \Phi(t')\| \leq C_\Phi |t - t'|^{1/2}, \quad (2)$$

where $\|\cdot\|$ is the Euclidean norm in R^2 .

The Lipschitz continuity (2) is stated above for a 2D case and it reads intuitively as a distance preserving property in the sense that points close to each other in the interval are transformed by Φ onto points close together in I_2 , but the converse is not necessarily true, since curve $\Phi(t)$, $t \in I_1$ intersects itself many times.

The next property will be useful for evaluating areas from samples along a space-filling curve.

Property 3 (F3 – measure preservation). *Space-filling curve Φ is the Lebesgue measure preserving in the sense that for every Borel $A \subset I_2$ we have $\mu_2(A) = \mu_1(\Phi^{-1}(A))$, where μ_1 and μ_2 denote the Lebesgue measure in R_1 and R_2 , respectively.*

At first glance, this property is strange. Note that it means that only values of lengths and areas before and after the transformation by Φ are equal. For example, an interval of the length 0.1 cm is transformed into a set having the area 0.1 cm^2 .

2.1.3 Quasi-inverses of space-filling curves

As mentioned above, points which are close in I_2 may have far, but not very far (see F2)) pre-images in I_1 . The reason is that Φ does not have the inverse Sagan (1994) in the usual sense

(intuitively, because a curve intersects itself). For our purposes it is of interest to find at least one $t \in I_1$ such that $\Phi(t) = x$ for given x . Consider a transformation $\Psi : I_2 \rightarrow I_1$, such that $\Psi(x) \in \Phi^{-1}(x)$, where $\Phi^{-1}(x)$ denotes the inverse image of x , i.e., the set $\{t \in I_1 : \Phi(t) = x\}$. Φ^{-1} allows to order linearly pixels in an image. We shall call Ψ a quasi-inverse of Φ .

Property 4 (F4 – Quasi-invers). *Let $\Phi : I_1 \xrightarrow{onto} I_d$ be a space-filling curve of the Hilbert, the Peano or the Sierpiński type. One can construct its quasi-inverse $\Psi : I_d \rightarrow I_1$ in such a way that it is also Lebesgue measure preserving.*

See Skubalska-Rafajłowicz (2004) for the constructive proof of this property.

2.1.4 Remarks on generating space-filling curves

It is important that there exist algorithms for calculating approximate value of the Peano, Hilbert and Sierpiński curves at a given point $t \in I_1$ with $O\left(\frac{d}{\varepsilon}\right)$ of arithmetic operations, where $\varepsilon > 0$ denotes the accuracy of approximation Butz (1971); Skubalska-Rafajłowicz (2003); Skubalska-Rafajłowicz (2001a)). Furthermore, quasi-inverses of these curves can also be calculated with the same computational complexity Skubalska-Rafajłowicz (2004); Skubalska-Rafajłowicz (2001b); Skubalska-Rafajłowicz (2001a)).

The specific self-similarities and the symmetries that space-filling curves usually possess, allow us to define a given space-filling curve. For example, consider Sierpiński's 2D curve. $\Phi(t) = (x(t), y(t))$ is uniquely defined by the following set of functional equations (see Sierpiński (1912) for the equivalent definition)

$$\begin{cases} x(t) = 1/2 - x(4t + 1/2)/2, \\ y(t) = 1/2 - y(4t + 1/2)/2 \\ 0 \leq t \leq 1/8, \\ x(t) = 1/2 + x(4(t - 7/8))/2, \\ y(t) = 1/2 - y(4(t - 7/8))/2 \\ 7/8 \leq t \leq 1, \\ x(t) = 1/2 + x(1 - 4(t - 1/8))/2, \\ y(t) = 1/2 - y(1 - 4(t - 1/8))/2 \\ 1/8 \leq t \leq 3/8, \\ x(t) = x(3/4 - t) \\ y(t) = 1 - y(3/4 - t) \\ 3/8 \leq t \leq 7/8. \end{cases} \quad (3)$$

It follows from (3) that $x(0) = y(0) = 0$ and $x(1/2) = y(1/2) = 1$. After above observation, one can convert (3) into recurrent algorithm of computing $\Phi(t)$, $t \in I_1$. If t has a finite binary expansion, $\Phi(t)$ is obtained in a finite number of iterations. The code for generating the Sierpiński space-filling curve is provided in the Appendix.

2.2 Equidistributed sequences in general

Equidistributed sequences are deterministic sequences, which behave like random variables, which are drawn from a uniform distribution, but they are much more regular. They arise as a tool for numerical integration, which is applied like the well known Monte-Carlo method, but provides much more accurate results, at least for carefully selected sequences.

Definition 2. A deterministic sequence $(x_i)_{i=1}^n$ is called equidistributed (EQD) (or uniformly distributed or quasi-random) sequence in I_d if

$$\lim_{n \rightarrow \infty} n^{-1} \sum_{i=1}^n g(x_i) = \int_{I_d} g(x) dx \quad (4)$$

holds for every continuous function g on I_d .

We refer the reader to Kuipers and Niederreiter (1974) for account on properties of EQD sequences and on their discrepancies, which are measures of their "uniformity". We shall use this definition mainly for $d = 1$ and $d = 2$, but the properties, which are proved below hold also for $d > 2$.

The well-known way of generating EQD sequences in $[0, 1]$ is as follows

$$t_i = \text{frac}(i\theta), \quad i = 1, 2, \dots, \quad (5)$$

where the fractional part is denoted as $\text{frac}(\cdot)$, θ is an irrational number.

A large number of methods for generating multivariate EQD sequences have been proposed in the literature, including generalizations of (5), Van der Corput sequences, Halton sequences and many others Davis and Rabinowitz (1984); Kuipers and Niederreiter (1974). As far as we know, none of them have properties which are needed for our purposes.

3. Generating sequences equidistributed along a space-filling curve

We propose a new class of equidistributed multidimensional sequences, which is obtained from one-dimensional equidistributed sequences by transforming it by a space-filling curve. In fact, one can combine any reasonable way of generating a one-dimensional EQD sequence with one of the space-filling curves of the Hilbert, Peano or Sierpiński type.

3.1 A new scheme of generating EQD sequences

The proposed scheme of generating an equidistributed sequence along a space-filling curve is as follows.

Step 1) Calculate t_i 's as in (5) (or as a one-dimensional Van der Corput sequence),

Step 2) Select one of the above space-filling curves as $\Phi : I_1 \rightarrow I_d$ and calculate x_i 's as follows:

$$x_i = \Phi(t_i), i = 1, 2, \dots, n. \quad (6)$$

For given n and θ it suffices to perform Steps 1) and 2) only once and store the resulting sequence $x_i, i = 1, 2, \dots, n$. An example is shown in Fig. 2.

Proposition 1. Sequence $\{x_i\}_{i=1}^n, x_i \in R^d$, which is generated according to the above method is the equidistributed sequence in I_d .

Proof. For continuous $g : I_d \rightarrow R$,

$$n^{-1} \sum_{i=1}^n g(x_i) = n^{-1} \sum_{i=1}^n g(\Phi(t_i)) \rightarrow \int_0^1 g(\Phi(t)) dt = \int_{I_d} g(x) dx, \quad (7)$$

since $\{t_i\}_{i=1}^n$ are EQD, Φ is continuous, while the last equality follows from F1).•

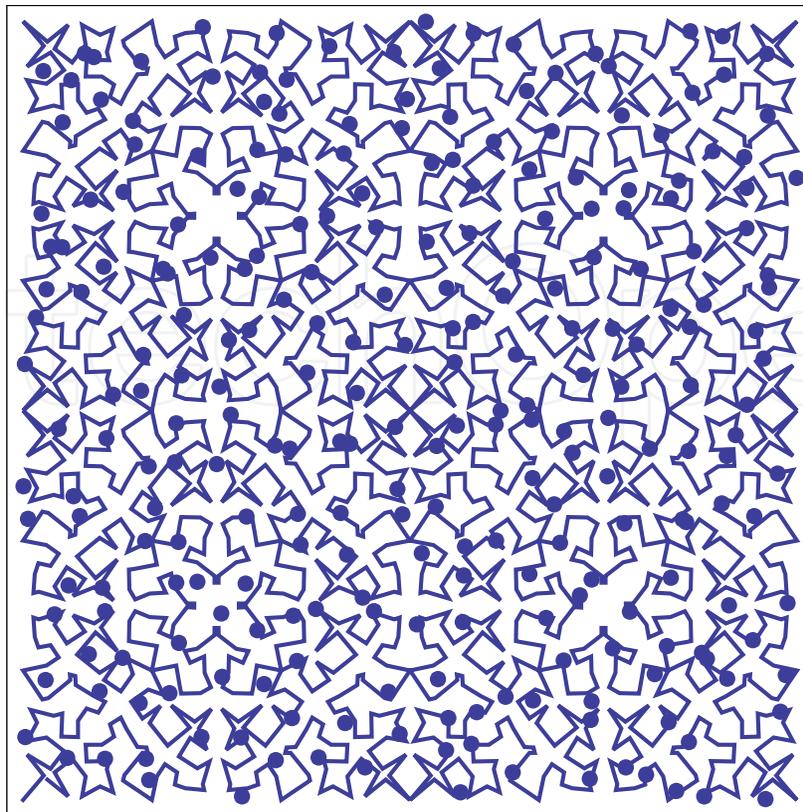


Fig. 2. The Sierpiński SFC and $n = 256$ EQD points.

3.2 Sampling of images

Application of the above sequence for sampling images is straightforward, but requires some preparation.

Preparation Perform Step 1 and Step 2, described in Section 3.1, for $d = 2$ in order to obtain EQD sequence $[x_i^{(1)}, x_i^{(1)}], i = 1, 2, \dots, n$.

Step 3 Scale and round sequence (6) as follows:

$$n_h(i) = \text{round}(N_h x_i^{(1)}), \quad n_v(i) = \text{round}(N_v x_i^{(1)}), \quad i = 1, 2, \dots, n, \quad (8)$$

where $[n_h(i), n_v(i)]$ denote coordinates of pixels in a real image, which has N_h pixels width and N_v pixels height.

Step 4 Read out samples $f_i = f([n_h(i), n_v(i)]), i = 1, 2, \dots, n$.

Remark 1. In practice, samples are collected as in Step 4 above, but for theoretical discussions we shall consider "theoretical" sample values $f_i = f(x_i), i = 1, 2, \dots, n$.

Remark 2. Note that gray levels f_i 's are usually stored as integers from 0 to 255, instead of $[0, 1]$, as it is assumed about f and f_i later on in this chapter.

4. Properties of the sampling scheme

This section is the central point of the chapter, since we collect here basic properties of the proposed sampling scheme. Some of them can be obtained by using known equidistributed

sequences, but properties presented in Section 4.3 and the reconstruction methods discussed in Section 5 essentially use unique features of sequences, which are equidistributed along a space-filling curve.

4.1 Images as measurable functions

Function $f : R^d \rightarrow R$ is called measurable if for every $c \in R$ the following level sets $\{x : f(x) < c\}$ are measurable (see, e.g., Wheeden and Zygmund (1977) for the definition). In this chapter we treat images f as measurable functions. This is convenient from a mathematical point of view. On the other hand, the class of measurable functions is sufficiently wide to include real life gray level images. This class, in particular, contains discontinuous functions, which can be expressed as limits of sequences of continuous functions. Furthermore, limits of such limits are also measurable functions and this process can be iterated, leading again to measurable functions.

Color images in RGB format can be modelled as triples of measurable functions.

4.2 Spectrum approximation

Denote by $\mathcal{F}(\omega)$, $\omega = [\omega^{(1)}, \omega^{(2)}]^T$ the Fourier transform of image f , i.e.,

$$\mathcal{F}(\omega) = \int_{I_2} \exp(-\mathbf{j} \omega^T x) f(x) dx, \quad (9)$$

where $\mathbf{j}^2 = -1$. We approximate spectrum \mathcal{F} by

$$\hat{\mathcal{F}}_n(\omega) = n^{-1} \sum_{i=1}^n \exp(-\mathbf{j} \omega^T x_i) f_i, \quad (10)$$

where x_i 's are EQD along a space-filling curve.

Proposition 2. *If f is measurable in I_2 and sampled at EDQ points along a space-filling curve, then for every ω we have*

$$\lim_{n \rightarrow \infty} |\mathcal{F}(\omega) - \hat{\mathcal{F}}_n(\omega)| = 0. \quad (11)$$

The proof of this property is deferred to the Appendix. Note that this result was obtained without assuming that f is band-limited. For earlier results in this direction see Unser (1998). A detailed discussion of the convergence rate of $\hat{\mathcal{F}}_n(\omega)$ to $\mathcal{F}(\omega)$ is outside the scope of this chapter, since it requires some smoothness assumptions imposed on f . We only mention that if for f the Lipschitz condition with the exponent $0 < \alpha \leq 1$ holds, i.e.,

$$|f(x') - f(x'')| \leq C_f \|x' - x''\|^\alpha,$$

where $C_f > 0$ is a constant, then

$$|\mathcal{F}(\omega) - \hat{\mathcal{F}}_n(\omega)| \leq C (\log(n)/n)^{\alpha/2},$$

where $C > 0$ is a constant, which may depend on f , the kind of a space-filling curve and ω , but not on n .

4.3 Fast approximate segmentation and blob analysis based on samples

The segmentation of images is a basic technique for marking objects, which are characterized by (approximately) the same gray level. In other words, our aim is to mark (approximately) regions such that

$$\{x \in I_2 : G_1 < f(x) < G_2\}, \quad (12)$$

where $0 \leq G_1 < G_2 \leq 1$ are specified thresholds. The next step is to find blobs, which are cluster of points, which are close to each other and far from points, which belong to another cluster. The segmentation and blob analysis task is time consuming, since it requires to not only visit each pixel and to mark it (as black, say), if $G_1 < f(x) < G_2$, but also to group marked into clusters, which usually requires to visit marked pixels several times (see, e.g., Davies (2005)).

We can reduce the computational burden by performing the segmentation and the blob analysis directly on samples. The blob analysis is also called silhouettes analysis, which are extracted by the segmentation.

Assume that sample points (t_i, f_i) , $i = 1, 2, \dots, n$ are reordered according to their first coordinates. Denote by $t_{(i)}$ i -th point of the equidistributed sequence in I_1 . Thus, after sorting $t_{(i)} < t_{(i+1)}$, $i = 1, 2, \dots, n$. Simultaneously – we keep the corresponding gray levels, which are denoted by $f_{(i)}$'s, i.e.,

$$f_{(i)} = f(\Phi(t_{(i)})), \quad i = 1, 2, \dots, n \quad (13)$$

Thus, our samples have the form $(t_{(i)}, f_{(i)})$. Now the procedure for approximate segmentation and blob analysis runs as follows.

Segmentation For each sample point mark $t_{(i)}$ as "black", if

$$G_1 < f_{(i)} < G_2 \quad i = 1, 2, \dots, n.$$

Blob analysis Starting from $t_{(1)}$, search for the first group of consecutive points

$$t_{(p)} < t_{(p+1)} < \dots < t_{(q)}, \quad 1 \leq p < q,$$

which are marked as "black". Then, repeat this search starting from $t_{(q+2)}$ ($t_{(q+1)}$ cannot be a member of the first group) and find the second group etc. Attach a label, e.g., number or color, to each group and treat it as the approximation of a blob.

Measuring blobs For each blob calculate the difference between the last point and the first point, i.e., $t_{(q)} - t_{(p)}$ and treat it as the approximation of the area of the corresponding blob.

The segmentation step does not require explanation (see Fig. 3). In the second step we use F2) property of space-filling curves that is if points $t_{(j)}$ and $t_{(j+1)}$ are close, the also points $x_{(j)} = \Phi(t_{(j)})$ and $x_{(j+1)} = \Phi(t_{(j+1)})$ are close in the image. To justify the last step, let us note that, according to F3) and F4), the length $|t_{(q)} - t_{(p)}|$ can be used as the approximation of the area of the smallest polygon containing $x_{(j)}$, $j = p, p + 1, \dots, q$.

The idea of the approximate blob analysis is illustrated in Fig. 3. The white, gray and black squares (left panel) were sampled in 512 points, which are equidistributed along the Sierpiński space-filling curve. The resulting gray levels are shown in the right panel. Note that samples from the white square are almost perfectly grouped as samples, which are numbered as 320 to 440. Similarly, samples from the black and light gray squares are grouped in the right panel as samples from 60 to (almost) 200 and from 200 to 320, respectively. Samples from the dark gray

square are split into two groups. The first one is numbered from 1 to 60. the second one, from 420 to 512. The consequence of this (unavoidable) split is not too severe, since we obtain two blobs of dark gray color (if $G_1 \approx 185$, $G_2 \approx 195$) instead of one, but when transformed to the image space, these two blobs will be close to each other. The only points, which would lead to false grouping are shown as separate points in the right panel of Fig. 3. This is the price paid for speeding up grouping. We can avoid even these false classifications by checking a proper classification of small clusters, but at the expense of an additional computational burden. The above approach can be applied to images in RGB format, just by applying it to each channel separately, but keeping the same sequence $t_{(i)}$'s.

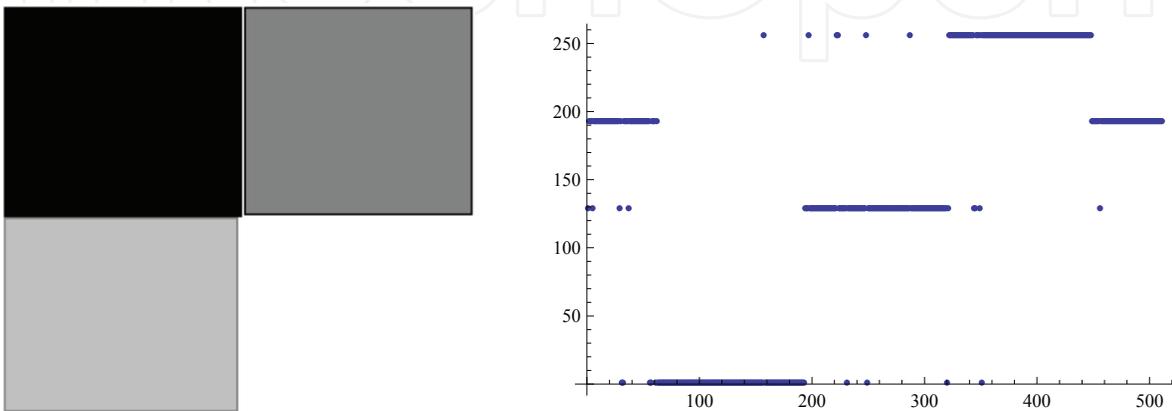


Fig. 3. Explanation why (approximate) segmentation and blob analysis work.

4.4 Approximating moments

Moments a_k of image f with respect to linearly independent functions $v_k(x)$ are defined as

$$a_k = \int_{I_2} f(x) v_k(x) dx, \quad k = 1, 2, \dots \quad (14)$$

a_k 's are usually approximated by the sums of gray levels located at all the pixels. We can gain much on efficiency using $\hat{a}_k^{(n)} = n^{-1} \sum_{i=1}^n f_i$ to evaluate theoretical moments from samples.

Proposition 3. Let f and v_k , $k = 1, 2, \dots$ be measurable functions in I_2 . Then, for f_i sampled at points x_i , which are equidistributed along a space-filling curve, we have

$$\lim_{n \rightarrow \infty} |a_k - \hat{a}_k^{(n)}| = 0, \quad (15)$$

i.e., approximate moments converge to the theoretical moments as the number of samples grows to infinity.

We omit the proof, since it is similar to the one for the spectrum approximation.

The role of moments in image analysis is well established (see, e.g., Davies (2005); Pawlak (2006)). In particular, selecting $v_k(x)$'s as ordered monomials one can evaluate centroids of blobs, their area etc. Central moments, in turn, provides translation invariant information about shape parameters describing blobs.

4.5 Moving mean and median filtering

The moving mean and the moving median are the most popular filters applied in image processing. A rectangular window of size $(2P + 1) \times (2Q + 1)$, say, is scanning the image and the mean value (or the empirical median) of gray levels of the corresponding pixels replaces the central pixel value.

Assuming that samples are ordered as in Section 4.3, one can perform (approximately) the same kind of filtering directly on samples. The filtering process runs as follows.

Step 1 Sort samples according to their first coordinates in order to obtain $(t_{(i)}, f_{(i)})$.

Step 2 Select half of the size of a neighborhood, which is used for filtering. Denote it by S .

Step 3 Starting from $i = S + 1$ to $i = n - (S + 1)$, perform the following operations:

1. calculate

$$\hat{f}_i = (2S + 1)^{-1} \sum_{m=-S}^S f_{(i-m)} \quad (16)$$

or the empirical median of the following gray levels

$$\{f_{(i-m)} : m = -S, \dots, 0, 1, \dots, S\},$$

2. \hat{f}_i (or by the empirical median) is attached to the point $t_{(i)}$ in the filtered sample.

As usual, we are faced with the boundary problem, since we cannot filter samples numbered by $i \leq S$ and $i \geq n - S$. The simplest remedy is to leave these samples unchanged.

Somewhat more sophisticated way of median filtering along a space-filling curve was proposed in Krzyżak (2001), but – in opposite to the present chapter – neighbors were not equidistributed.

As is known, sampling of images and space-filling curves have many other applications (see, e.g., Davies (2001); Lamarque and Robert (1996)), in which the sampling scheme proposed here can also be useful.

5. Approximate reconstruction by k-nearest neighbors RBF nets

Our aim is to demonstrate that images can be efficiently reconstructed from the samples, which are equidistributed along a space-filling curve. We shall concentrate on reconstruction schemes, which are based on nearest neighbors and artificial neural networks from the radial basis functions (RBF) class.

An alternative way would be to estimate the spectrum of an image according to (10) on a regular grid and to calculate the inverse discrete Fourier transform by the FFT algorithm.

5.1 Reconstruction using RBF nets and exact neighbors

Consider $N_h \times N_v$ image. The coordinates of its pixels are denoted as (h, v) , while positions of sample points are denoted as $(n_h(i), n_v(i))$, $i = 1, 2, \dots, n$. Abusing the notation, we shall write $f_{(k,m)}$, forgetting for a while that earlier f was defined in $[0, 1]^2$.

5.1.1 1-NN reconstruction scheme

A seemingly naive algorithm of reconstructing the underlying image is the following.

Preparations For all $N_h N_v$ positions (h, v) pixels find the nearest neighbor (1-NN) among positions of samples $(n_h(i), n_v(i))$, $i = 1, 2, \dots, n$ and store these positions in $N_h \times N_v$ table C , say. Its elements $c(h, v)$, $h = 1, 2, \dots, N_h$, $v = 1, 2, \dots, N_v$ contain addresses to the closest sample point.

Step 1 Repeat Step 2 for $h = 1, 2, \dots, N_h$, $v = 1, 2, \dots, N_v$.

Step 3 Attach gray level of the nearest sample point $f_{c(h,v)}$ to pixel (h, v) .

The most time consuming Step 1 is performed only once. As a result we obtain table C , which is of the same size as an original image and – at a first glance – all the compression effect is distracted. Note however, that when frames from a long video sequence are sampled and later some of them have to be reconstructed, then it pays to store matrix C in order to have an almost immediate reconstruction of selected frames. This is exactly the case when a production quality is monitored by a camera and we have to store (and keep for a long time) very long sequences of images, which document the quality of products.

5.1.2 k-NN reconstruction using RBF net

We can generalize the above reconstruction scheme by taking into account gray levels of nearest neighbors starting from the first one, second nearest up to k -th nearest. It is convenient to express such a generalized reconstruction scheme as a neural network from the well known radial basis functions (RBF) class.

To this end, we select a nonnegative kernel $K : R_1 \rightarrow R_1$, which is a function such that

$$\int_{-\infty}^{\infty} t K(t) dt = 0, \quad \int_{-\infty}^{\infty} t^2 K(t) dt < \infty, \quad (17)$$

which is normalized $K(0) = 1$. This kind of normalization is not typical, but convenient for our purposes. Typical examples include the uniform kernel ($K(t) = 1$, $|t| < 1$ and zero otherwise), the Epanechnikov kernel etc.

Denote by $\tilde{f}_{(h,v)}$ the reconstructed gray level at (h, v) , which is calculated as follows

$$\tilde{f}_{(h,v)} = \sum_{j=1}^k w_j(h, v) f_{c(j,h,v)}, \quad (18)$$

where $c(j, h, v)$ is the address of j -th closest point among positions of samples $(n_h(i), n_v(i))$, $i = 1, 2, \dots, n$, while weights $w_j(h, v)$ are defined as follows:

$$w_j(h, v) = \frac{K(\|(h, v) - c(j, h, v)\|^2 / H(k))}{\sum_{j=1}^k K(\|(h, v) - c(j, h, v)\|^2 / H(k))}, \quad (19)$$

where

$$H(k) \stackrel{def}{=} \|(h, v) - c(k, h, v)\|^2. \quad (20)$$

Note that when $k = 1$ and kernel K is the uniform one, then (18) reduces to 1-NN reconstruction scheme.

We remark that (18) is the approximation scheme rather than interpolatory one, as it was used in Anton et al (2001).

5.2 Reconstruction using RBF nets and neighbors along SFC

We can reduce the computational burden on finding nearest neighbors by replacing the exact search by the approximate one, which is performed along a space-filling curve. The proposed method is as follows.

1-NN along SFC

Step 1 For all pixels (h, v) perform the following steps:

1. normalize current pixel (h, v) to I_2 as $x_{h,v} \stackrel{def}{=} (h/N_h, v/N_v)$.
2. calculate its quasi-inverse $t_{hv} \stackrel{def}{=} \Psi(x_{h,v})$
3. find its nearest neighbor among all $t_{(i)}$'s and denote its number by $\hat{c}(h, v)$.

and store the resulting $N_h \times N_v$ matrix as \hat{C} .

Step 2 As the approximate value of f at pixel (h, v) (or at $x_{h,v}$) take f at $\hat{c}(h, v)$.

Step 3 Repeat Step 2 for all (h, v) .

The main advantage of this scheme is in that finding NN among ordered $t_{(i)}$'s has computational complexity $O(\log_2(n))$. The price for that is a possibility of missing the true NN in I_2 , since in Step 1 we use the quasi-inverse of SFC. Nevertheless, a point found in this is close to NN in I_2 due to F2). Matrix \hat{C} can be treated as approximation of matrix C in the sense that many of its entries are the same as the corresponding entries of matrix C . The differences arise due to self-crossing of SFC.

We do not provide details of reconstruction by RBF net, which is based on approximate nearest neighbors, since changes in (19) and (20) are obvious.

5.3 Reconstruction by local random spreading of grey levels

In opposite to the above-described reconstruction schemes, which are based on searching (approximate) neighbors to each pixel, the method considered here spreads gray levels of samples in their neighborhoods. Below, we describe the simplest way of such spreading, which is based on a random choice of neighbors.

Reconstruction by random spreading

Step 1 Prepare $N_h \times N_v$ matrix S , say, as follows. Fill its entries, denoted as $s(h, v)$ by sampled gray levels at appropriate positions. The remaining entries fill by "empty" symbol (coded as a number greater than 1 (or 255)).

Step 2 Check whether "empty" entries are present in S . If not, the stop and S contains the reconstructed image. Otherwise, go to Step 3.

3 Find the position of the next "empty" element of matrix S and denote it by (h, v) .

Step 4 Select at random (with equal probabilities) one of the following directions "up", "down", "left", "right".

Step 5 Assign the contents of $s(h-1, v)$ to $s(h, v)$, if the direction is "left". Assign the contents of $s(h+1, v)$ to $s(h, v)$, if the direction is "right" etc. Go to Step 2.

In Step 5 it may happen that the contents assigned to $s(h, v)$ is still "empty", but after a short time gray levels of samples nicely "smear" over the image. The result of the reconstruction is random, but repeated reconstructions produce visually stable images in a relatively short time. In Steps 4 and 5 one can use the neighborhood containing eight or more pixels.

5.4 Examples

As explained in the Introduction, the proposed sampling and reconstruction schemes are dedicated mainly for industrial images. However, it is instructive to verify their performance using the well-known example, which is shown in Fig. 4. Analysis of the differences between the original and the reconstructed images indicate that 1-NN reconstruction scheme provides the most exact reconstruction, but the reconstruction by random spreading provides the nicest looking image.

The application to industrial images is illustrated in Fig. 5, in which a copper slab with defects is shown. Note that it suffices to store 4096 samples in order to reconstruct 1000×1000 image, without distorting gray levels of samples from the original image. This is equivalent to the compression ratio of about $1/250$. Such a compression rate plus loss-less compression allows us to store a video sequence (30 fps) from one month of a continuous production process on a disk or tape, having 1 TB (terra byte) capacity.

6. Appendix – proof of Proposition 3

Take arbitrary $\epsilon > 0$. By the Lusin theorem, there exists a set $E = E(\epsilon/4)$ such that $f|_E$ is continuous and $\mu_2(E - I_2) < \epsilon/4$. Denote by $\mathcal{F}_E(\omega)$ the Fourier transform of $f|_E$. Then, for $D \stackrel{\text{def}}{=} E - I_2$ we have

$$|\mathcal{F}(\omega) - \mathcal{F}_E(\omega)| = \left| \int_D e^{-j\omega^T x} f(x) dx \right| < \mu_2(D) < \frac{\epsilon}{4}, \quad (21)$$

since both integrands do not exceed 1. Let

$$\hat{\mathcal{F}}_E(\omega) = n^{-1} \sum_{x_i \in E} \exp(-j\omega^T x_i) f_i. \quad (22)$$

Define $\Delta_n = |\hat{\mathcal{F}}_n(\omega) - \hat{\mathcal{F}}_E(\omega)|$. Then

$$\Delta_n = \left| n^{-1} \sum_{x_i \notin E} \exp(-j\omega^T x_i) f_i \right|. \quad (23)$$

Clearly, $\Delta_n \leq \mathcal{N}(I_2 - E)/n$, where

$$\mathcal{N}(I_2 - E) \stackrel{\text{def}}{=} \text{card}\{i : x_i \in (I_2 - E)\}.$$

From Proposition 1 it follows that for $n \rightarrow \infty$

$$\Delta_n \leq \frac{\mathcal{N}(I_2 - E)}{n} \rightarrow \mu_2(I_2 - E) < \epsilon/4. \quad (24)$$

Thus, for n sufficiently large we have $\Delta_n < \epsilon/4$. Define

$$\delta_n = \left| \left(\frac{1}{n} - \frac{1}{\mathcal{N}(E)} \right) \sum_{x_i \in E} \exp(-j\omega^T x_i) f_i \right|$$

where $\mathcal{N}(E) \stackrel{\text{def}}{=} \text{card}\{i : x_i \in E\}$. Clearly,

$$\left| \sum_{x_i \in E} \exp(-j\omega^T x_i) f_i \right| \leq \mathcal{N}(E).$$

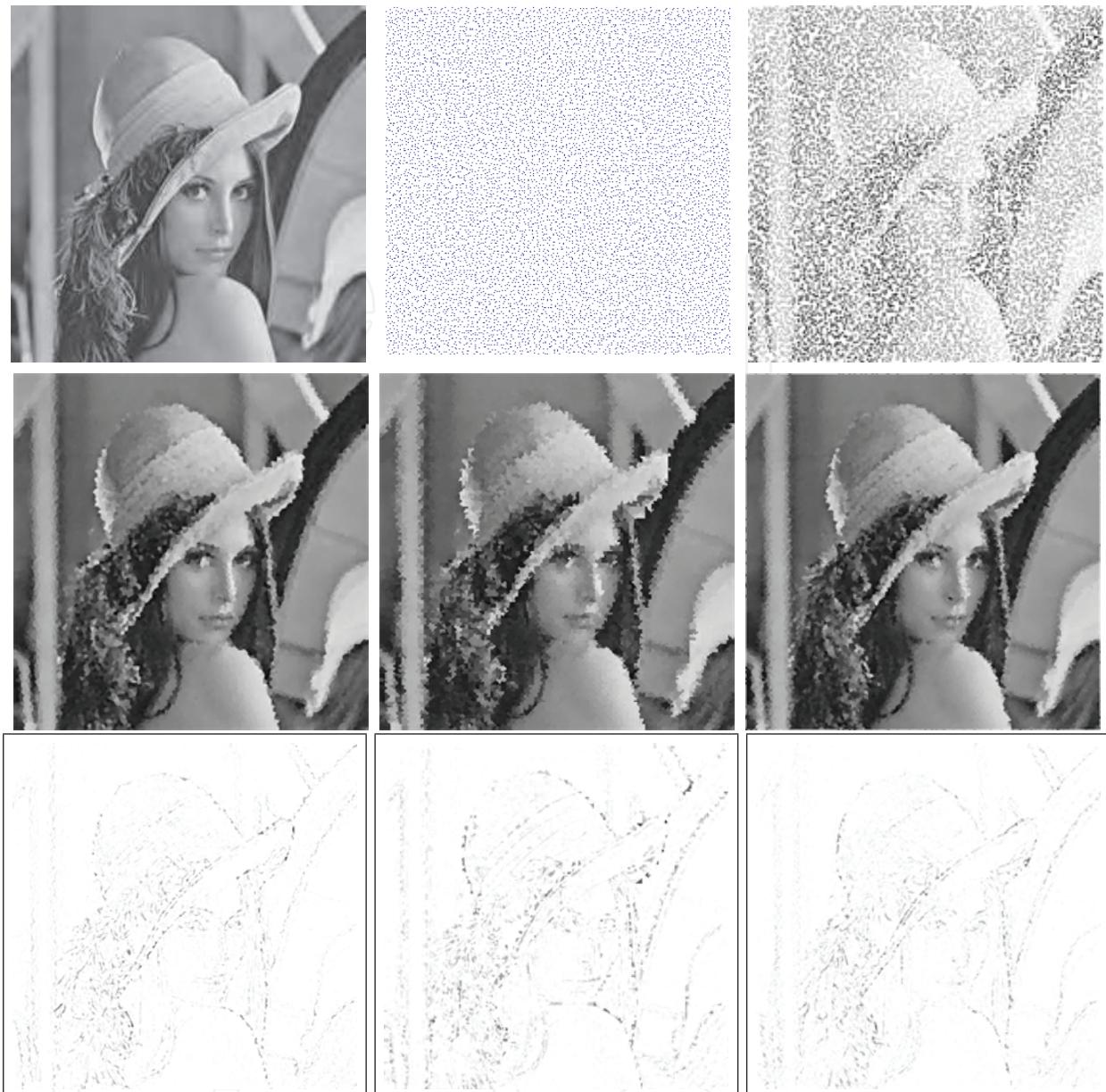


Fig. 4. Lena image, 512×512 pixels, (upper-left panel) sampled at 10 000 points equidistributed along the Sierpiński space-filling curve (upper-middle panel). Gray levels at sample points are shown in the upper-right panel. The results of reconstruction by 1-NN method (middle left panel), by 1-NN along the space-filling curve (central panel) and by spread to random-NN (middle right panel). The differences between the original image and the reconstructed one are shown in the last row of this figure.

Thus, for n large enough

$$\delta_n \leq |(\mathcal{N}(E)/n - 1)| < \epsilon/4, \quad (25)$$

since, by Proposition 1, $|(\mathcal{N}(E)/n - \mu_2(I_2))| \rightarrow 0$ as $n \rightarrow \infty$. We omit argument ω in the formulas that follow. Summarizing, we obtain.

$$|\mathcal{F} - \hat{\mathcal{F}}_n| < \epsilon/4 + |F_E - \hat{\mathcal{F}}_n|, \quad (26)$$

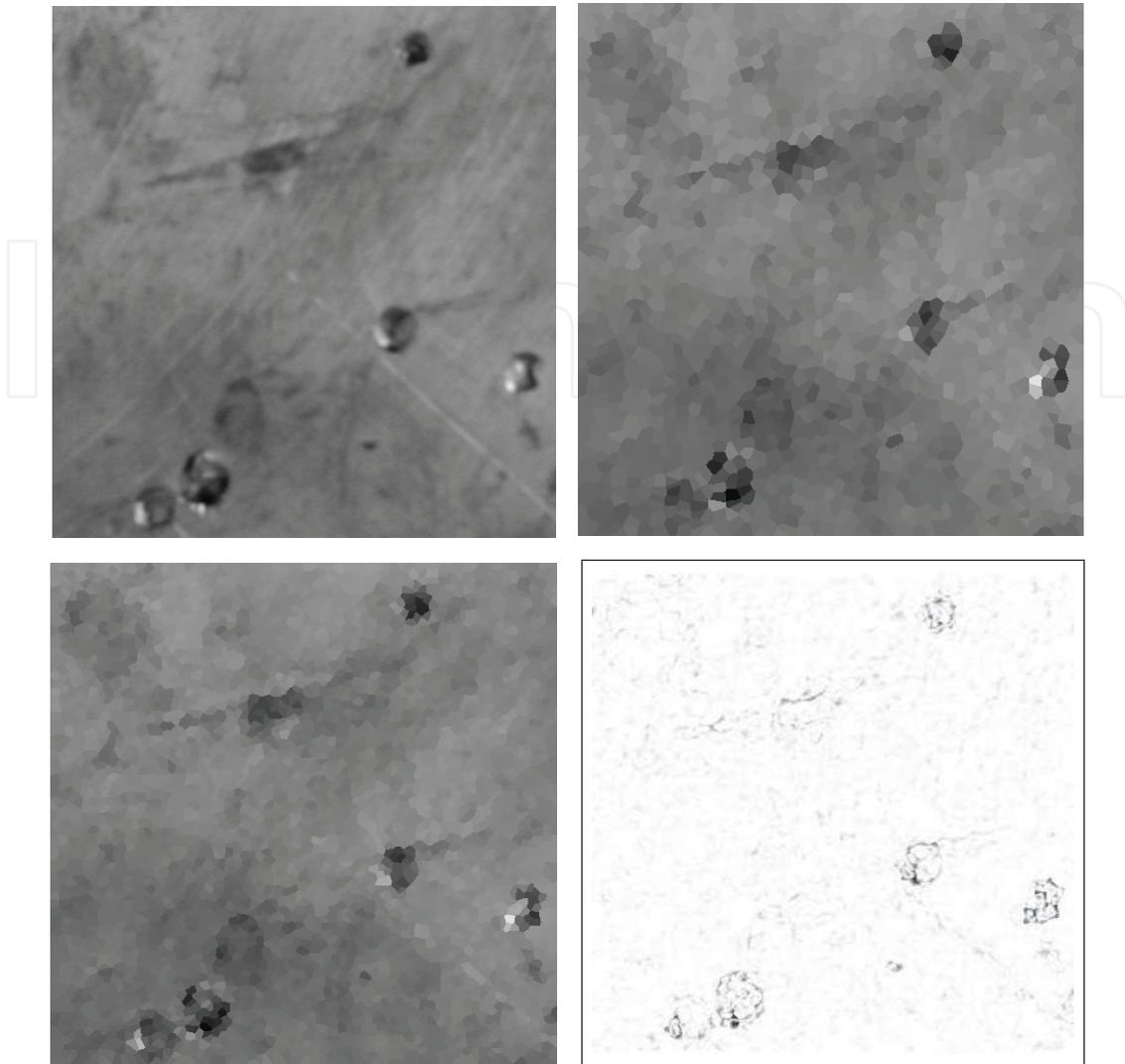


Fig. 5. Copper slab with defects, 1000×1000 pixels (upper left panel) and its reconstruction from $n = 2048$ samples by 1-NN method (upper right panel). The same slab reconstructed from $n = 4096$ samples (lower left panel) and the difference between the original image and the reconstructed one (lower right panel). Compression ratio $1/250$.

since, by (21), $|\mathcal{F} - \mathcal{F}_E| < \epsilon/4$. Analogously,

$$|\mathcal{F}_E - \hat{\mathcal{F}}_n| < \epsilon/4 + |\hat{\mathcal{F}}_E - \hat{\mathcal{F}}_n|, \quad (27)$$

due to (24). Finally,

$$|\mathcal{F}_E - \hat{\mathcal{F}}_E| \leq \delta_n + \left| \mathcal{F}_E - \frac{1}{\mathcal{N}(E)} \sum_{x_i \in E} \exp(-j \omega^T x_i) f_i \right|. \quad (28)$$

The last term in (28) approaches zero, since f is continuous in E and Proposition 1 holds. Hence, $|\mathcal{F}_E - \hat{\mathcal{F}}_E| < \epsilon/2$ for n large enough, due to (25). Using this inequality in (27) and invoking (26) we obtain that for n large enough we have $|\mathcal{F} - \hat{\mathcal{F}}_n| < \epsilon$. •

7. Appendix – Generating the Sierpiński space-filling curve and equidistributed points along it.

In this Appendix we provide implementations of procedures for generating points from the Sierpiński space-filling curve and its quasi-inverse, which are written in Wolfram's Mathematica language. Special features of new versions of Mathematica are not implemented with the hope that the code should run and be useful for all versions, starting from version 3.

The following procedure `tranr` calculates one point of the Sierpiński curve, i.e., for given $t \in I_1$ an approximation to $\Phi(t) \in I_d$ is provided, but only for $d \geq 2$ and even. Parameter k of this procedure controls the accuracy to which the curve is approximated. It should be a positive integer. In the examples presented in this chapter $k = 32$ was used.

```
tranr[d_,k_,t_] := Module[{bd,cd,ii,j,jj,tt,KM,km,be,kb},
bd=1; tt:=t;xx={1};
Do[bd=2^ii-bd+1; AppendTo[xx,1],{ii,d-1}];
cd=bd*2^(-d); km={};
Do[kb=Floor[(tt-cd/2^d)*2^d]+1;
tt=2^d*(tt-cd/2^d-(kb-1)*2^(-d));
If[kb==2^d, kb=0];
If[Floor[kb/2]<kb/2,tt=1-tt]; AppendTo[km,kb] ,{j,k}];
Do[ KM=km[[k-j+1]]; ww={};
Do[ If[KM< 2^(d-jj),be=0,be=1]; AppendTo[ww,be];
KM=KM-be*2^(d-jj);
If[be==1,KM=2^(d-jj)-KM-1] ,{jj,d}];
Do[xx[[d-jj+1]]=1/2-(1/2-ww[[jj]])*xx[[d-jj+1]] ,{jj,d}]{j,k}];
(*out*) xx]
```

The following lines of the Mathematica code generate the sequence of 2D points, which are equidistributed along the Sierpiński space-filling curve.

```
dim = 2; deep = 32; n = 512; th = (Sqrt[5.] - 1.)/2.; {i, 1, n}];
points = Map[tranr[dim, deep, #] &, Sort[Table[FractionalPart[i*th]]];
```

8. References

- Anton F.; Mioc D. & Fournier A. (2001) Reconstructing 2D images with natural neighbour interpolation. *The Visual Computer*, Vol. 17, No. 1, (2001) pp. 134-146, ISSN: 0178-2789
- Butz A. (1971) Alternative Algorithm for Hilbert's Space-filling Curve. *IEEE Trans. on Computing*, Vol. C-20, No. 4, (1971) pp. 424-426, ISSN: 0018-9340
- Cohen A.; Merhav N. & Weissman T. (2007) Scanning and sequential decision making for multidimensional data Part I: The noiseless case. *IEEE Trans. Information Theory*, Vol. 53, No. 9, (2007) pp. 3001-3020, ISSN: 0018-9448
- Davies, E.R. (2001) A sampling approach to ultra-fast object location. *Real-Time Imaging*, Vol. 7, No. 4, pp. 339-355, ISSN: 1077-2014
- Davies, E.R. (2005) *Machine Vision*, Morgan Kaufmann, ISBN: 0-12-206093-8, San Francisco
- Davis P. & Rabinowitz P. (1984) *Methods of Numerical Integration*, Academic Press, ISBN: 0-12-206360-0, Orlando FL
- Kamata S.; Niimi M. & Kawaguchi, E. (1996) A gray image compression using a Hilbert scan. *Proceedings of the 13th International Conference on Pattern Recognition*, Vienna, Austria, August, 1996, Vol. 3, pp. 905-909

- Krzyżak A.; Rafajłowicz E. & Skubalska-Rafajłowicz E. (2001) Clipped median and space-filling curves in image filtering. *Nonlinear Analysis: Theory, Methods and Applications*, Vol. 47, No. 1, pp 303-314, ISSN: 0362-546X
- Kuipers L. & Niederreiter H. (1974) *Uniform Distribution of Sequences*. Wiley, ISBN: 0471510459/9780471510451, New York
- Lamarque C. -H. & Robert F. (1996) Image analysis using space-filling curves and 1D wavelet bases, *Pattern Recognition*, Vol. 29, No. 8, August 1996, pp 1309-1322, ISSN: 0031-3203
- Lempel, A. & Ziv, J. (1986) Compression of two-dimensional data. *IEEE Transactions on Information Theory*, Vol. 32, No. 1, January 1986, pp. 2-8, ISSN: 0018-9448
- Milne S. C. (1980) Peano curves and smoothness of functions. *Advances in Mathematics*, Vol. 35, No. 2, 1980, pp. 129-157, ISSN: 0001-8708
- Moore E.H. (1900) On certain crinkly curves. *Trans. Amer. Math. Soc.*, Vol. 1, 1900, pp. 72-90
- Pawlak M. (2006) *Image Analysis by Moments*, Wrocław University of Technology Press, ISBN: 83-7085-966-6, Wrocław
- Platzman L.K. & Bartholdi J.J. (1989) Spacefilling curves and the planar traveling salesman problem. *Journal of the ACM*, Vol. 36, No. 4, October 1989, pp. 719-737, ISSN: 0004-5411
- Rafajłowicz E. & Schwabe R. (2003) Equidistributed designs in nonparametric regression. *Statistica Sinica*, Vol. 13, No 1, 2003, pp. 129-142, ISSN: 1017-0405
- Rafajłowicz E. & Skubalska-Rafajłowicz E. (2003) RBF nets based on equidistributed points. *Proceedings of the 9th IEEE International Conference on Methods and Models in Automation and Robotics MMAR 2003*, Vol. 2, pp. 921-926, ISBN: 83-88764-82-9, Międzyzdroje, August 2003
- Rafajłowicz E. & Schwabe R. (1997) Halton and Hammersley sequences in multivariate non-parametric regression. *Statistics and Probability Letters*, Vol. 76, No. 8, 2006, pp. 803-812, ISSN: 0167-71-52
- Regazzoni, C.S. & Teschioni, A. (1997) A new approach to vector median filtering based on space filling curves. *IEEE Transactions on Image Processing*, Vol. 6, No, 7, 1997, pp. 1025-1037, ISSN: 1057-7149
- Sagan H. (1994) *Space-filling Curves*, Springer ISBN: 0-387-94265-3, New York
- Schuster, G.M. & Katsaggelos, A.K. (1997) A video compression scheme with optimal bit allocation among segmentation, motion, and residual error. *IEEE Transactions on Image Processing*, Vol. 6, No. 11, November 1997, pp. 1487-1502, ISSN: 1057-7149
- Sierpiński W. (1912) Sur une nouvelle courbe continue qui remplit toute une aire plane. *Bull. de l'Acad. des Sci. de Cracovie A.*, 1912, pp. 463-478
- Skubalska-Rafajłowicz E. (2001a) Pattern recognition algorithms based on space-filling curves and orthogonal expansions. *IEEE Trans. Information Theory*, Vol. 47, No. 5, 2001, pp. 1915-1927, ISSN: 0018-9448
- Skubalska-Rafajłowicz E. (2001b) Data compression for pattern recognition based on space-filling curve pseudo-inverse mapping. *Nonlinear Analysis: Theory, Methods and Applications* Vol. 47, No. 1, (2001), pp. 315-326, ISSN: 0362-546X
- Skubalska-Rafajłowicz Ewa. (2003) Neural networks with orthogonal activation function approximating space-filling curves. *Proc. 9th IEEE Int. Conf. Methods and Models in Automation and Robotics. MMAR 2003*, Vol. 2, pp. 927-934, ISBN: 83-88764-82-9, Międzyzdroje, August 2003,

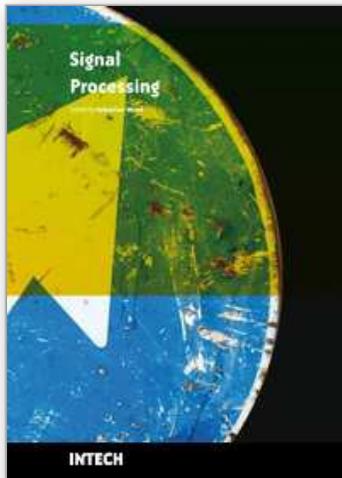
- Skubalska-Rafajłowicz E. (2004) Recurrent network structure for computing quasi-inverses of the Sierpiński space-filling curves. *Lect. Notes in Comp. Sci.*, Springer 2004, Vol. 3070, pp. 272–277, ISSN: 0302-9743
- Thevenaz P.; Bierlaire M. & Unser M. (2008) Halton Sampling for Image Registration Based on Mutual Information, *Sampling Theory in Signal and Image Processing*, Vol. 7, No. 2, 2008, pp. 141-171, ISSN: 1530-6429
- Unser M. & Zerubia J. (1998) A generalized sampling theory without band-limiting constraints, *IEEE Trans. Circ. Systems II*, Vol. 45, No. 8, 1998, pp. 959-969, ISSN: 1057-7130
- Wheeden R. & Zygmund A. (1977) *Measure and Integral*, Marcell Dekker, ISBN: 0-8247-6499-4, New York
- Zhang Y. (1997) Adaptive ordered dither. *Graphical Models and Image Processing*, Vol. 59, No. 1, January 1997, pp. 49-53, ISSN: 1077-3169
- Zhang Y. (1998) Space-filling curve ordered dither. *Computers and Graphics*, Vol. 22, No 4, August 1998, pp 559-563, ISSN: 0097-84-93
- Zhang Y. & Webber R. E. (1993) Space diffusion: an improved parallel halftoning technique using space-filling curves. *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pp 305-312, ISBN: 0-89791-601-8, Anaheim, CA, August 1993

Acknowledgements This work was supported by a grant contract 2006-2009, funded by the Polish Ministry for Science and Higher Education.

IntechOpen

IntechOpen

IntechOpen



Signal Processing

Edited by Sebastian Miron

ISBN 978-953-7619-91-6

Hard cover, 528 pages

Publisher InTech

Published online 01, March, 2010

Published in print edition March, 2010

This book intends to provide highlights of the current research in signal processing area and to offer a snapshot of the recent advances in this field. This work is mainly destined to researchers in the signal processing related areas but it is also accessible to anyone with a scientific background desiring to have an up-to-date overview of this domain. The twenty-five chapters present methodological advances and recent applications of signal processing algorithms in various domains as telecommunications, array processing, biology, cryptography, image and speech processing. The methodologies illustrated in this book, such as sparse signal recovery, are hot topics in the signal processing community at this moment. The editor would like to thank all the authors for their excellent contributions in different areas of signal processing and hopes that this book will be of valuable help to the readers.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Ewa Skubalska-Rafajlowicz and Ewaryst Rafajlowicz (2010). Space-Filling Curves in Generating Equidistributed Sequences and Their Properties in Sampling of Images, Signal Processing, Sebastian Miron (Ed.), ISBN: 978-953-7619-91-6, InTech, Available from: <http://www.intechopen.com/books/signal-processing/space-filling-curves-in-generating-equidistributed-sequences-and-their-properties-in-sampling-of-ima>

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2010 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](#), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen